



LIBRARY OF THE  
UNIVERSITY OF ILLINOIS  
AT URBANA-CHAMPAIGN

510.84  
Il6r  
no. 361-366  
cop 2



The person charging this material is responsible for its return to the library from which it was withdrawn on or before the **Latest Date** stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

To renew call Telephone Center, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

MAR 5 1980

MAR 8 REC'D



ILLIAC IV  
QUARTERLY PROGRESS REPORT  
July, August, and September 1969

Contract No.  
USAF 30(602)-4144

JAN 13 1970

THE LIBRARY OF THE  
JAN 14 1970  
UNIVERSITY OF ILLINOIS

ILLIAC IV Doc. No. 231



DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS



Digitized by the Internet Archive  
in 2013

<http://archive.org/details/illiacivquarterl362univ>

ILLIAC IV  
QUARTERLY PROGRESS REPORT  
July, August, and September 1969

Contract No.  
USAF 30(602)-4144

Department of Computer Science  
University of Illinois at Urbana-Champaign  
Urbana, Illinois  
61801

November 1, 1969

This work was supported in part by the Department of Computer Science, University of Illinois, Urbana, Illinois, and in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center, under Contract No. USAF 30(602)-4144.





# TABLE OF CONTENTS

	Page
REPORT SUMMARY . . . . .	1
1. HARDWARE . . . . .	3
1.1 Logic Simulation and Diagnostics . . . . .	3
1.1.1 Logic Simulation . . . . .	3
1.1.1.1 PE Simulator . . . . .	3
1.1.1.2 CU Simulator System . . . . .	3
1.1.2 Card Test Generation . . . . .	3
1.1.2.1 Card Test Generator . . . . .	3
1.1.2.2 Card Test Translation . . . . .	4
1.1.2.3 Generation and Evaluation of Card Tests . . . . .	4
1.1.3 PE Diagnostics . . . . .	5
1.1.4 PEX Computer and Supervisor System . . . . .	5
1.2 Design Automation . . . . .	6
1.2.1 Post Processor Program . . . . .	6
1.2.2 CU Printed Circuit Layout . . . . .	6
2. SOFTWARE . . . . .	7
2.1 Operating System . . . . .	7
2.2 Translator Writing System . . . . .	8
2.3 Compilers and Translators . . . . .	8
2.3.1 TRANQUIL . . . . .	8
2.3.2 GLYPNIR . . . . .	9
2.4 Assembler . . . . .	9
2.5 Utilities . . . . .	9
2.6 B5500 Operation . . . . .	10
3. APPLICATIONS . . . . .	11
3.1 Numerical Analysis . . . . .	11
3.1.1 Monte Carlo Transport . . . . .	11
3.1.2 Mathematical Subroutines . . . . .	11
3.1.3 Automatic Solution of Initial Value Problems . . . . .	12
3.1.4 Eigenvalue Problems . . . . .	13
3.2 Linear Programming . . . . .	13
3.3 Long Codes . . . . .	14
3.4 Radar Data Processing . . . . .	15
3.5 Seismic Signal Processing . . . . .	15
3.6 Conversion of Burroughs ALGOL to CDC 6000 Series ALGOL . . . . .	16
3.7 ILLIAC IV Education and Documentation . . . . .	16
3.7.1 CS 491-D . . . . .	16
3.7.2 Training Programs . . . . .	17
3.7.3 Documentation . . . . .	18
3.7.4 ILLIAC IV Textbook . . . . .	18
3.8 Graphics . . . . .	19
REFERENCES, THESES, AND DOCUMENTS . . . . .	21



## REPORT SUMMARY

At the conclusion of this quarter the following milestones were agreed upon by the University of Illinois and the Burroughs Corporation:

<u>Event</u>	<u>Date</u>
Debugged Prototype PE	Oct. '69
Debugged Production PE	Feb. '70
CU Ready for Integration, Start System Integration	Feb. '70
System Confidence Demonstration	Apr. '70
Ship One-Quadrant System	Aug. '70
Final Acceptance Test at University of Illinois	Nov. '70

The semi-conductor memories to be produced by Fairchild are two weeks to one month behind schedule. The memory chips for the prototype processing element memory appear to be sound and operable, however, Fairchild is experiencing yield problems. Fairchild is currently developing a new set of procedures and equipment to test memory chips at the wafer level.

Texas Instruments is currently meeting Burroughs' delivery schedule in providing the dual-in-line packaged ECL circuits for ILLIAC IV.

The PDP-9 computer, which will be attached to the PE exerciser to provide the ability to automatically diagnose errors in the PE and CU boards, was delivered to the University in August. Components for the interface to the PE exerciser have arrived at the University and the interface is currently under construction. Software development for the PDP-9 is proceeding on schedule.

A new version of the "ILLIAC IV Systems Characteristics and Programming Manual" has been delivered by Burroughs to the University

and distributed to the appropriate personnel. The University and Burroughs are working together to gather comments, criticisms and corrections to be incorporated into the next update of the manual to be published in January, 1970.

FOURUM, the ILLIAC IV user's group, continues to grow and the present membership is at 140. The distribution for other research documents is in the process of being automated and all mailing lists are being centralized into one file.

Burroughs has agreed to accept an incremental conversion of the present contract to a fixed price contract. Target dates for specific tasks which are to be converted to fixed price are currently under negotiation.

The project initiated last quarter to permit Burroughs' ALGOL to be compiled and executed on a Control Data Corporation 6000 series machine has been partially completed. A preprocessing program that translates Burroughs ALGOL and XALGOL into CDC ALGOL is currently operational. A more comprehensive program should be available next quarter.

The building plans for the Center for Advanced Computation were approved by the President of the University of Illinois and the Board of Trustees. Bids for general construction have been let.

## 1. HARDWARE

### 1.1 Logic Simulation and Diagnostics

#### 1.1.1 Logic Simulation

##### 1.1.1.1 PE Simulator

The CU Card Simulator Generator System has been extended and applied to the generation of PE Card Simulators. The net lists (wire lists) of 35 PE card types were compiled from Burroughs' logic diagrams. Error messages from the programs of the Generator System helped in removing some errors in translation and in card punching of the net list as well as in the logic diagrams. The connector pin assignment in the diagrams was compared with Burroughs' assignment list which was being used in the generation of the backplane wire list. All of the PE card net lists have been released for simulation and test generation.

##### 1.1.1.2 CU Simulator System

The CU Card Simulator Generator System is now complete, with the facilities for automatic sequential execution and history generation.

Some experiments of CU Card Simulation were made and the system was verified.

The system will be used for debugging the logic of CU cards and for generating Bad Card Simulators.

The design of the CU Section Simulator Generator System was resumed in September. The System will be implemented in the next quarter and used for checking the design of the Control Unit.

#### 1.1.2 Card Test Generation

##### 1.1.2.1 Card Test Generator

In this period, a final version of the Card Test Generation System has been completed and debugged.

To allow more flexibility and to meet additional requirements (e.g., initialization of the latches present in some boards and holding some signal values at a constant during the simulation instead of varying them in a random manner), a new program has been incorporated into the system.

It is expected that the capabilities of the system will be extended in the near future to facilitate failure isolation in addition to failure detection.

#### 1.1.2.2 Card Test Translation

The two main functions of the Card Test Translator are: (1) to print out the test patterns and the list of failures associated with patterns in an adequate form, and (2) to translate the patterns and failure information into PEXTAP (PE Exerciser Test Assembler) language.

The first function has been completed and the second function is being debugged. The program will be completed in the next quarter with some extension of the functions.

#### 1.1.2.3 Generation and Evaluation of Card Tests

More than half of the 35 PE cards have been simulated for test generation. In most cases, a set of patterns which can detect all failures in the DIL packages on the card, was obtained in a short running time (between 10 seconds and 2 minutes) on the B5500. The rest of the time required on the computer was about 10 minutes including simulator compilation and random pattern generation.

Also, a few CU cards were simulated for test generation during this period. All failures were detected on one of the cards. The input partitioning and the initialization of the storage states seem to be very important for efficient test generation.

Burroughs has compiled a list of possible failure modes in DIL's by analysis of the circuits. Work is continuing at the University to correlate their results to the failure modes assumed in the Card Test



Generator. It appears that practically all failure modes defined by Burroughs can be covered as single or double failures in the Card Test Generator.

### 1.1.3 PE Diagnostics

The path test generator system was used to generate a revised test program for the new prototype PE. Combinational tests have not yet been revised for the new design.

One of the last programs of the Control Logic Test Generator System is being coded. This program will detect and solve conflicts in the assignment of values to input signals.

### 1.1.4 PEX Computer and Supervisor System

Delivery of a Digital Equipment Corporation PDP-9/L computer to be used as a PEX controller was made at the end of August. Characteristics of this system are:

- 1.5  $\mu$ sec cycle time
- 18 bit word length
- 16K words of core storage
- Hardware multiply and divide
- 3 DECTape transports
- 1 IBM compatible magnetic tape transport

Coding and assembly of the previously defined PEX supervisor is progressing satisfactorily. Early hands-on experience with the computer was gained through coding and assembling of several utility routines to speed development of the PEX supervisor. Typical of these is a magnetic tape handler which permits listings to be written on a magnetic tape whose contents are then printed using the high speed line printer of the Burroughs B5500.

The interface to the PEX will be made by modifying the second paper tape reader channel on the PEX, and then constructing an interface to the PDP-9 which will accept data words from the PDP-9's data channel

and serialize them into characters. Design of the interface was accomplished during August. A misunderstanding between Burroughs' and the University's logic designers about the exact details of the interface required that small modifications be made in mid-September.

Actual construction of the interface has been delayed awaiting arrival of the necessary components. It is expected that the interface construction will be completed at approximately the same time as the programming, and that the entire system will be delivered to Burroughs in time to check out the first production PE's.

## 1.2 Design Automation

### 1.2.1 Post Processor Program

The Post Processor (Printed Circuit Wiring Analysis) Program is being used for production of the CU printed circuit boards. The program's output has been useful as an additional tool for updating and completing the PC Artwork. The need for substantial manpower to manually check the automatically produced artwork has been eliminated.

### 1.2.2 CU Printed Circuit Layout

The University and Burroughs have completed arrangements for doing the DA for the CU PC layout. Approximately 12 hours of processing is done on the University's B5500 system per PC board. The University's system can provide better turnaround and throughput for this larger amount of processing and save additional billing to the project. It is planned to complete this effort during the next quarter.



## 2. SOFTWARE

### 2.1 Operating System

Several modifications were made to the operating system to provide for the first pass October system. A second job parser was designed and is being debugged. This job parser is a simpler module than the first and should be more reliable during early debugging of the first system. The loader functions have been subsumed by the program collector and execution monitor. Since multiple relocation and overlay loading would not be required in the first system, the program collector was modified to produce a relocated program segment suitable for direct loading into the ILLIAC IV memory. Thus, the loader was made unnecessary and the read operation needed to place the program in the array was assigned to the execution monitor.

The hardware supervisor intrinsics were frozen and the implementation of buffer files on the B6500 has begun. Contrary to expectations, Burroughs will not include the buffer file facility in its first Master Control Program (MCP).

The execution monitor, data processor, and job partner depend heavily on the MCP. Due to the moving target nature of the MCP, the specification of critical sections of each of these modules was not made until mid-September. It was decided to freeze the design on the basis of current knowledge of the MCP and to modify the delivered MCP to represent the situation on which the design was frozen. It is expected that this may entail an extensive amount of debugging time when the B6500 is first delivered.

Parts of the system have been coded in SIMULA so that their activity can be simulated and debugged before the arrival of the B6500. Currently, simulation activity is concerned with disk allocation and quadrant execution.

## 2.2 Translator Writing System

Most of the steps of the system described last time have been implemented and effectively debugged. BNF2FPL/TWS, FPL2PAR/TWS, PAR2ALG/TWS, and the parser files are all functional. TWINKLE/DISK is working, except for a few of the TWINKLE constructs; work on ISL/DISK is just beginning. Use of the new system has already begun. From the results obtained so far, the expectations for the new system have been met. The interpretive parser is very similar in all respects to the previous one, but the executable parser is significantly smaller. The executable parser is small enough (13K core estimate for executable TESLA compiler compared with 10K for the interpretive version and 18K or 20K for the old executable version) that it has become a very desirable choice for small to medium-sized languages. A further advantage is that three different pieces, the filling of the parsing tables, the lookahead procedure, and the Floyd production procedure calls, can be made executable or interpretive independently.

## 2.3 Compilers and Translators

### 2.3.1 TRANQUIL

Progress on the TRANQUIL compiler was slowed during this quarter due to personnel changes and vacations. Brisk activity on the compiler was resumed during the last two weeks of the quarter. While coding and debugging of the TRANQUIL basic subset due for completion in October continues, development and design effort has been centered on implementation of explicit I/O statements for the language.

The implementation of procedures in TRANQUIL has been completed on time. Some slippages are apparent in the coding and debugging of the basic subset. It was hoped to have coding completed by the end of October, but November is now the target date. Debugging will take longer and will be a function, in part, of the user's perseverance in utilizing what is available.

### 2.3.2 GLYPNIR

Work on GLYPNIR proceeded slowly during this quarter because of a reduction in staff. Even so, several additions and extensions to the compiler were made. The compiler now accepts numeric literals, including floating point numbers with exponents and base designators. Also, partial word operators (similar to those of BURROUGHS EXTENDED ALGOL), the AS construct (which allows the programmer to make a GLYPNIR register synonymous with an ILLIAC IV hardware register), and several new control card options have been added which will make debugging easier.

Work is also progressing on the debugging of system subroutines such as sine, cosine, tangent, natural log, and square root. These system subroutines will be introduced during the early part of the next quarter.

### 2.4 Assembler

July and August were spent in fixing the final design for the macro assembler. Implementation of the macro assembler, according to this design, began in September.

### 2.5 Utilities

The utilities effort commenced during this quarter. The initial effort is concentrated on developing two formatted data processing programs. One will take data from external sources and put it on the B6500 disk in the form required by the operating system for transmission to the ILLIAC IV disk. The other program performs the inverse transformation: it takes a file which was transmitted to the B6500 disk from the ILLIAC IV disk and formats the data for external representation.

Other areas under consideration are:

- (a) debugging facilities
- (b) graphic display software
- (c) software to permit user measurement of program performance with an eye to hardware usage and time optimization.

## 2.6 B5500 Operation

The present status of the MCP is Mark IX, LEVEL 51.14. In July 175.16 hours, in August 174.70 hours and in September 202.39 hours of processor time was logged.

### 3. APPLICATIONS

#### 3.1 Numerical Analysis

##### 3.1.1 Monte Carlo Transport

A two-dimensional Monte Carlo Transport program is currently being written in GLYPNIR. Straight storage is used for the variable storage scheme. The program will accommodate approximately  $64 \times 64$  spacial cells, and only  $180 \times 64$  particles, but appropriate buffers are being provided so that particles can be streamed in and out from the disc. The coding is more than 50 percent complete.

##### 3.1.2 Mathematical Subroutines

The following subroutines have been completed and simulated.

<u>Function</u>	<u>Precision in Octal Digits</u>
sine	15 ~ 16
cosine	15 ~ 16
tangent	15 ~ 16
cotangent	15 ~ 16
arccosine	15 ~ 16
arcsine	15 ~ 16
arctangent	15 ~ 16
natural logarithm	14 ~ 16
exponential	14 ~ 16
square root	15 ~ 16

The subroutines are entered with 64 arguments and they return 64 answers. Computer Approximations, by Hart [1], was the source for the approximating functions. Accuracy was determined by comparing results from the ILLIAC IV simulator with double precision results from the 360/75. Corresponding routines for 32 bit operands will be provided as they are needed.

### 3.1.3 Automatic Solution of Initial Value Problems

An investigation was initiated this quarter to determine the feasibility of constructing a general purpose program to solve initial value problems of the form:

$$\frac{\partial \vec{u}}{\partial t} = \vec{A}(\vec{u}) \quad (1)$$

where  $A$  may be a non-linear operator involving partial derivatives with respect to spacial variables. Especially encouraging in this investigation is the success the University has had with automatic integration programs to solve large sets of ordinary differential equations of the form:

$$\frac{d\vec{z}}{dt} = \vec{f}(\vec{z}) \quad (2)$$

It can be shown for certain operators  $A$  (for example  $\partial^2/\partial x^2$ ) that many of the successful finite difference schemes for solving (1) may be derived by first discretizing the spacial variable, forming a large set of ordinary differential equations, and then solving the ordinary differential equations using a conventional scheme. Considering accuracy and stability there is no difference in the approach. This suggests that we may capitalize on the success of the automatic integration of ordinary differential equations in the integration of partial differential equations.

The two pressing questions which must be investigated are:

- (a) Can the spacial mesh be refined independently if the algorithm for solving the ordinary differential equations has certain properties?
- and (b) Will it be possible to first approximate  $A$  by another operator  $A^*$  which will insure that discontinuities in the solution (i.e. shocks) will not completely destroy the numerical solution? If these questions can be answered affirmatively, it may then be practical to develop an automatic solution program.



### 3.1.4 Eigenvalue Problems

The program, for Jacobi's method for eigenvalues of symmetric matrices with a "Triangular Skew" storage scheme, has been written and is now in the process of being simulated and debugged.

The problem most apparent at the present time is finding a suitable method of keeping track of the elements in a most efficient way. Since in each transformation of the matrix only those row-elements change their locations whose index  $n \leq \#$  of transformations, the method described in QPR Document #219 [2] seems inefficient, since it calculates all locations of all elements.

The method being developed sees the location of the elements of one row as a function of the locations of the elements of the previous row with the adjustment that each row has one element less than the previous one. This method also involves a minimum of element-manipulation.

### 3.2 Linear Programming

Major projects for the Linear Programming (LP) group this quarter have been the SSK simulation of LPS (small linear programming system) and an investigation of the gradient projection method as a technique for LPL (large linear programming system).

One of the significant benefits that linear programming by the simplex method will receive from ILLIAC IV is in the realm of "multiple pricing". Behind this concept lies the fact that no computer-implemented LP system can afford to examine the entirety of a large constraint matrix (A-matrix) each iteration. Typically, reduced prices ( $d_j$ 's) are computed for only one portion of the A-matrix columns (partial pricing). A smaller number (about 5), with the most favorable prices, are selected for pivot evaluations. This is multiple pricing. Updating such a set of vectors is time-consuming, so suboptimization is performed with this set; that is, as many of these vectors as possible are introduced into the basis before reconsidering the remaining A-matrix columns.

The massive "crunching" capability of the ILLIAC IV allows the entire A-matrix to be scanned in determining the pricing vectors. A major iteration consists of pricing all columns of the A-matrix and selecting and transforming those selected for multiple pricing. LPS handles up to 127 (vs. 1-10 on a conventional machine) of these vectors for suboptimization. The suboptimization steps are called minor iterations. The pivot element is determined for each of these transformed vectors; the one producing the greatest improvement in the objective function is selected for pivoting. When the remaining transformed vectors produce insufficient objective function improvement, a new major iteration is initiated.

The pricing and candidate selection portion of the major iteration (DSCHUZ) has been successfully simulated this quarter. Consisting of three sequentially executed steps, DSCHUZ (1) calculates the  $d_j$ 's for the entire A-matrix and selects an optimum subset--usually 512--of these, (2) refines this set to  $N \leq 127$  vectors and (3) executes an inter-PE sort on the selected values. The result is a column-sorted array containing tags and cost values for the selected vectors. Then the N most probable vectors to enter the basis can be column-sequentially processed in minor iterations. Simulation tests of other LPS routines will be carried out as machine time becomes available.

Investigation of gradient projection as a solution technique for LPL (large linear programming system) has produced a preliminary recursion algorithm for the method. It takes a form which is similar in concept and importance to the recursion introduced by the product form inverse in place of the explicit inverse of the revised simplex method. Studies of the vector densities produced and the resulting storage requirements are somewhat alarming at present; modifications are being considered as the investigation continues.

### 3.3 Long Codes

During this quarter, the identification of a stationary dynamical system with plant and observation noises was attempted. A



least-squares estimation yielded an estimation error that remains bounded, but it did not provide an upper bound on the size of this error in the limit.

A stochastic approximation algorithm was constructed based on some theorems by Dvoretzky [3] and by Venter [4]. It was shown that the proposed algorithm yields an estimation error that approaches zero in the mean-square sense.

### 3.4 Radar Data Processing

During this period, work was completed on the documentation of the ILLIAC IV Kalman Filter. A complete write-up has been printed and is available as ILLIAC IV Document #222, entitled "A Kalman-Filter Tracking Program for ILLIAC IV" [5].

A complete subroutine package for doing Fourier transforms is being written for ILLIAC IV. It uses the Cooley-Tukey algorithm and can handle data sets which can vary from 8 to 4096. The algorithm is programmed in a 32-bit floating point mode for complex numbers, where the real and imaginary value of each reside in one 64-bit PE. This program is expected to be running by the next quarter and complete documentation will be started.

### 3.5 Seismic Signal Processing

Several programs, previously written, were modified to make them more flexible. One research assistant, working on these programs, was beginning to convert these B5500 ALGOL programs to ILLIAC IV assembly language until the draft board assigned him to a different area of responsibility.

Research into multi-channel filter design has continued during this quarter.

Since the data received in seismic recording is all real, rather than complex, a modification of the Fast Fourier Transform

program to accept two streams of real data has started. This program has been postponed until the fourth quarter when additional personnel will be acquired.

### 3.6 Conversion of Burroughs ALGOL to CDC 6000 Series ALGOL

A preprocessing program that translates Burroughs ALGOL and XALGOL into CDC ALGOL is operational. Both, the ILLIAC IV assembler and simulator, have been properly processed by this program.

Certain Burroughs constructs, such as SCAN and REPLACE statements, are changed into function calls. All references to input-output are also changed into function calls. The necessary subroutines must be written for the CDC machine. It is expected that work should be completely finished by the end of November 1969.

Programs will run 20 percent faster on the CDC machine than previously estimated.

### 3.7 ILLIAC IV Education and Documentation

#### 3.7.1 CS 491-D

The graduate course in Computer Science "Architecture, Applications and Languages for a Parallel Computer" is being offered again this fall. The course outline is as follows:

#### I. Machine Design . . . (Walt Heimerdinger) . . . . 6 classes

Conventional Machines.....1

Unconventional Machines....5

Holland/SOLOMON...1

ILLIAC IV.....3

ILLIAC IV/B6500...1

A short history of conventional machine design will preface a discussion of the antecedents of ILLIAC IV, namely the Holland and SOLOMON machines. The discussion of the ILLIAC IV organization will include discussion of the architecture of the B6500.

II. Operating System and I/O . . (Marv Graham) . . . 2 classes  
(Tom Mason )

The operating system is presented as the "glue" binding the B6500 and the ILLIAC IV. The I/O subsystem will be explored for timing and operational relationships.

III (a). Introduction and General  
Overview of Software . . (Prof. Northcote) . . 1 class

III (b). Software . . . . . 11 classes

Assembler/Simulator . . (Dave Grothe ) . . 3  
(Gary Grossman)  
GLYPNIR . . . . . (Duncan Lawrie) . . 4  
TRANQUIL . . . . . (Norma Abel ) . . 4

The major languages of the ILLIAC IV are presented. An effort will be made to highlight the useful aspects.

IV. Communicating with the User . . (Pete Alsberg) . . 1 class

A discussion of data storage devices and data displays.

V. Applications . . . . . 8 classes

Numeric . . . . . (Dave McIntyre ) . . . 4  
Radar . . . . . (Morris Knapp ) . . . 2  
Linear Programming . . (Serena Yardley) . . . 2

The major uses of these applications will be sketched in terms of how ILLIAC IV is essential to these areas. Techniques dependent on unique characteristics of ILLIAC IV will be emphasized.

### 3.7.2 Training Programs

As the delivery of ILLIAC IV draws closer, it will be necessary to initiate training programs to educate people in the use of ILLIAC IV. Various types of programs are currently under consideration to service three types of people.

- (a) Personnel working directly for the ILLIAC IV Project at the University of Illinois (U of I)
- (b) Personnel at the U of I who do not work directly for the ILLIAC IV Project
- (c) Personnel outside the U of I who do not work directly for the ILLIAC IV Project but are interested in becoming users of ILLIAC IV.

### 3.7.3 Documentation

Due to the growing list of people, both internal and external to the U of I, who utilize ILLIAC IV Reports and Documents and the User's Group--FOURUM, a master file has been established. This master file will effectively automate and simplify mailing procedures and will centralize user descriptive data into one source.

The Library of ILLIAC IV documents is currently undergoing re-organization to allow for future expansion and to make it easier to use. ACM (Association for Computing Machinery) category codes will be coded to existing and future documents.

### 3.7.4 ILLIAC IV Textbook

Work has started this quarter on "The ILLIAC IV Computer--A Programmer's Text".

The book is an attempt to present, under one cover, the information necessary to instruct a programmer in the use of ILLIAC IV. To this end, the book has been partitioned into two distinct sections. The first section, the chapters, are in conventional textbook format including problems to be solved by the student and topics for discussion. To facilitate the learning process, the information presented in the chapters is quite general in nature--only those concepts necessary to give the reader a basic understanding of ILLIAC IV are presented.

The second section, the appendices, refine the information presented in the chapters. The appendices act as a reference manual, or a handbook. Either section of the book can be read independently of the other.

Chapter I is a review--a selected history of computing machines. The evolution of digital computers is described in terms of the problems that had to be solved. The growing, changing capabilities of computers are discussed, leading finally to ILLIAC IV.

Chapter II is a broad outline of the ILLIAC IV hardware structure. A sample problem (LaPlace's equation for temperature

distribution in two dimensions) is used to illustrate how the components of the hardware structure are tied together.

Chapter III describes the major programming languages available to the ILLIAC IV user. The sample problem of Chapter II is re-visited to explore the use of the assembly language ASK and two high level languages, GLYPNIR and TRANQUIL.

Chapter IV presents the system software that is available to the user. The I/O system including the B6500 computer, ILLIAC IV disk, DFC, BIOM, IOS, CDC and TMU is described. The operating system is covered in a skeletal fashion and a typical job is traced as it flows through the system. Software capabilities of specific hardware components such as remote terminals, graphics, IMP and the ARPA network, and various other peripheral equipments are also covered.

Chapter V discusses applications of ILLIAC IV and indicates the types of problems that are especially amenable to solution on such a parallel processor.

### 3.8 Graphics

The following specifications are currently being proposed as the graphic requirements for ILLIAC IV:

#### I. CRT Requirements

- A. 2048 X 2048 resolution with character and vector capabilities.
- B. Input channel of  $\geq 10^5$  bits/sec. (see II.B.)
- C. Need at least one CRT of TV size for viewing, with keyboard for limited interaction. "Light pen" interaction and intermediate hard copy with fair quality and limited quantity (~15 sec/page) are highly desirable if not too costly.

#### II. Photographic System

- A. 16 mm and 35 mm microfilm cameras with sufficient registration for movies.
- B. Required rate for microfilm estimated at 1 frame/sec with an estimated 5000 instruc/frame or 5000 characters/frame (this

requires CRT input channel of  $\geq 10^5$  bits/sec). Development time of less than 30 minutes desired.

- C. Final hard copy from microfilm should be of high quality and produced at high rate.

Vendors are currently being contacted to submit bids on equipment which will meet the above specifications.



## REFERENCES

- [1] Hart, J. F., et al. Computer Approximations. New York: Wiley and Sons, 1968.
- [2] ILLIAC IV Quarterly Progress Report, January, February, and March, 1969. DCS Report No. 339, ILLIAC IV Document No. 219. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.
- [3] Dvoretzky, A. "On Stochastic Approximation." Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, I. 1965, pp. 39-55.
- [4] Venter, J. H. "On Dvoretzky Stochastic Approximation Theorems." Annals of Mathematics and Statistics, Vol. 37, 1966, pp. 1534-1544.
- [5] Stevens, James E. "A Kalman-Filter Tracking Program for ILLIAC IV." ILLIAC IV Document No. 222, DCS File No. 804, (August 13, 1969).

## THESES

- Allegre, Nicole. "TESLA, A Control Language for Logic Simulations of Digital Circuits." Master's thesis, DCS Report No. 347. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.
- Flowerdew, Stanley J. "An Analysis of Some Eulerian Methods for One Dimensional, Inviscid, Compressible Dynamics." Master's thesis, DCS Report No. 352. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.
- Lermit, Raymond J. "An Error Analysis of Solutions to Sparse Linear Programming Problems." Master's thesis, DCS Report No. 351. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.
- Kraska, Paul W. "Array Storage Allocation." DCS Report No. 344. (August 1, 1969).

## DOCUMENTS

- Beals, Alan J., et al. "The Automatic Generation of Floyd Production Syntactic Analyzers." DCS Report No. 350. (September 9, 1969).

- Bernhard, Winfried H. "A Matrix Inversion Program for ILLIAC IV." ILLIAC IV Document No. 229, DCS File No. 814, (July 8, 1969).
- Knowles, Michael. "A Shadow Algorithm for Computer Graphics." ILLIAC IV Document No. 227, DCS File No. 811, (September 22, 1969).
- Marceau, I. W., et al. "Linear Programming on ILLIAC IV." ILLIAC IV Document No. 225, DCS File No. 808, (September 1, 1969).
- Matsushita, Y. "Hidden Lines Elimination for A Rotating Object." ILLIAC IV Document No. 220, DCS File No. 801, (July 2, 1969).
- Stevens, James E. "A Kalman-Filter Tracking Program for ILLIAC IV." ILLIAC IV Document No. 222, DCS File No. 804, (August 13, 1969).
- Wang, Paul J. "Romberg's Method in Numerical Quadrature." ILLIAC IV Document No. 223, DCS File No. 806, (July 1, 1969).
- Yasui, Toshio. "Programming the Tillotson Equation of State on ILLIAC IV." ILLIAC IV Document No. 224, DCS File No. 807, (August 20, 1969).

#### CORRECTIONS

The following theses and documents were not mentioned in previous QPR's:

- Gaffney, John L., Jr. "TACOS: A Table Driven Compiler-Compiler System." Master's thesis, DCS Report No. 325. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.
- Gold, David E. "A Model for Linear Programming Optimization of I/O-Bound Programs." Master's thesis, DCS Report No. 340. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.
- Graham, Marvin L. "The Role of Input/Output in a Computer System." ILLIAC IV Document No. 221, DCS File No. 800, (June 27, 1969).
- Heimerdinger, W. L. "String Processing on a Parallel Computer." DCS Report No. 299, (January 13, 1969).
- Horsman, Larry. "Design Considerations for Multilayer Transmission Line Circuit Boards." DCS Report No. 254, (January 2, 1968).



UNCLASSIFIED

Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Department of Computer Science University of Illinois Urbana, Illinois 61801		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	
		2b. GROUP	
3. REPORT TITLE ILLIAC IV QUARTERLY PROGRESS REPORT July, August, September 1969			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) July-September, 1969 - Progress Report of the ILLIAC IV Project			
5. AUTHOR(S) (First name, middle initial, last name)			
6. REPORT DATE November 1, 1969		7a. TOTAL NO. OF PAGES 26	7b. NO. OF REFS 5
8. CONTRACT OR GRANT NO. 46-26-15-305		9a. ORIGINATOR'S REPORT NUMBER(S) ILLIAC IV Document No. 231 DCS Report No. 362	
9. PROJECT NO. USAF 30(602)-4144		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC TR	
10. DISTRIBUTION STATEMENT Qualified requesters may obtain copies of this report from DCS.			
11. SUPPLEMENTARY NOTES NONE		12. SPONSORING MILITARY ACTIVITY Rome Air Development Center Griffis Air Force Base Rome, New York 13440	
13. ABSTRACT See the Report Summary on Page 1 within the Report itself.			

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
PE Simulator						
CU Simulator						
PE Diagnostics						
PEX Computer						
Design Automation						
Operating System						
Translator Writing System						
Compilers and Translators						
TRANQUIL						
GLYPNIR						
Assembler						
Utilities						
B5500 Operation						
Numerical Analysis						
Initial Value Problems						
Eigenvalue Problems						
Linear Programming						
Long Codes						
Radar Data Processing						
Seismic Signal Processing						
Burroughs ALGOL to CDC ALGOL Conversion						
Education and Documentation						
Graphics						













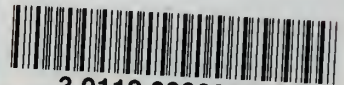




FEB 7 - 1974



UNIVERSITY OF ILLINOIS-URBANA  
510.84 IL6R no. C002 no.361-366(1969  
Digital computer internal report /



3 0112 088398877